

AIS Programming Standards

Powerbuilder Guidelines

- [Introduction](#)
- [Procedure](#)
- [Application Development](#)
- [Related Policies, Standards, & Guidelines](#)

Introduction

The following are the initial standards and general guidelines that should be deployed by application teams as they develop PowerBuilder client/server applications using SYBASE stored procedures. The guidelines listed are considered essential for providing a sound development and production environment.

Procedure

Application Development Environment

Directory Structures

Distinct network directory structures are mandatory in keeping the various PowerBuilder objects separated for application development and production deployment. Application directories suggested are listed below:

- **Working/Development:** Used for developing and modifying all PowerBuilder Objects.
- **Stage:** Used for combining all PowerBuilder objects, including vendor code with UCLA programs, for testing before actual move to production.
- **Production:** Used to keep the most current version of the client application. This is the application that is currently running in all user sites. Separate sub-directories should be maintained by date to keep backup runtime versions of the applications that may be called upon if a new release is not operational in production.
- **Vendor:** Used to keep copies of application as received from the vendor. It is suggested that the vendor directories be maintained by issue date in separate sub-directories.

Application Deployment Topics

- **PowerBuilder Client Application:** The client tested and approved version of the application from the “staging” directory gets sent to all user sites and waits to be deployed synchronized with database objects. Review that the application ‘ini’ file has been updated with any new requirements. The new application should not be officially deployed until notified by the Application Distributor that the Stored Procedures have been moved and compiled into the production database. Warning messages issued upon start up of the application should be considered as a critical component that warns users when they are currently executing a version of the client software that does not synchronize with the production database objects.
- **Sybase Stored Procedures/database objects:** When most user sites have received the new version of the PowerBuilder client application, then the Database objects, Stored procedures,

tables, etc. get compiled/created in the production database environment overnight while the users are off the system.

- **Sybase/PowerBuilder DLLs:** All necessary and up to date Sybase/PowerBuilder dll's, icons, and bitmaps must be distributed with the client application to all user sites. This includes the current Sybase Open Client dlls that permit the connection to the Sybase database.

Application Development Standards

Standardization of various components of the PowerBuilder application are mandatory in providing a consistent, easy-to-use, and maintainable system.

- **Class Library:** This is a key factor in maintaining a uniform look and feel for any PowerBuilder application. This could be either developed internally or purchased from a third party.
- **Inheritance:** Inheritance should be used from class library objects as much as possible. This will minimize redundant code for widely used objects and functionality's. Also, the application will maintain a uniform appearance.
- **Object Naming Conventions:** Each name uniquely names the object. You need to use a meaningful name for each object. Some generic names like dw_1 or dw_2 are acceptable in ancestor objects for uniformity and ease of referencing. When a vendor product is used, the UCLA object names should be prefixed with a site letter or identifier.

Object Type Prefixes:

General form: {objecttype prefix}_{name}

Example: wf_open_sheet

UCLA Example: uwf_open_sheet

wf	Window function
ws	Window structure
ms	Menu Structure
uf	User object function
us	User object structure
dw	Data window control

Variable Names: Variable names are composed of three parts.

General form: {datatype}{scope}_{name}

Examples:

ib_inquiry	a instance variable of type boolean
ls_lastname	a local variable of type string

Part 1 - Data Type:

w	window
m	menuitem
dw	datawindow
str	structure
uo	user object
i	integer
ui	unsigned integer
l	long
ul	unsigned long
b	boolean
s	string
db	double

r real
c decimal (used for currency)
d date
t time
dt datetime

Part 2 - Scope:

g global
s shared
i instance
l local

Part 3 - Name: The variable name. It needs to be descriptive. Local variables used for counters such as i_x are acceptable as long as there are a reasonable number of them in the script (usually around three).

- **Script/Stored Procedure Documentation:** All scripts and stored procedures should be well documented for ease of maintenance. The following information is mandatory for all Sybase stored procedures and PowerBuilder scripts:
 - - Description of Purpose: Describe the name of the object and the functionality.
 -
 - Change Log - Reason, Description, Date, Programmer: keep a log of the creation and maintenance of the object for future reference. This will help others maintaining the object.
- **GUI Screen Design Considerations:** All application windows should be consistent in design and on-line presentation. The following are guidelines that should be utilized when building PowerBuilder applications.
 - All window fonts, buttons, radio buttons, and check boxes should be consistent from one window to the next within the application. See the following examples:
 - Arial 8 point font, non bold, should be used for all standard inquiry windows.
 - Inquiry windows should be displayed as flat screens.
 - Input fields in Maintenance Screens should be displayed as 3 D lower boxes.
 - Use a bold font for editable fields it and labels describing all fields should be defined as regular, not bold.
 - Screen Navigation - The screen should flow from top to bottom and left to right.
 - Function Keys should be defined consistently throughout the application.
 - Windows and all objects on them (buttons, text, descriptions, labels, etc.) should be tested on a test workstation that has been configured to match a broad range of user population environment to insure a complete and proper display.
 - All date and monetary fields should be displayed in a consistent manner throughout the application.
- **PowerBuilder to Sybase Interface/Data Retrieval Guidelines**
 - **Stored Procedures:** Should be used for all complex datawindows. This will make it easy to modify the logic in the future if business rules change. This approach basically eliminates the redistribution of the client application in most cases because the changes are made on the server.
 - **PB Scripts:** Should be used for quick and simple retrieval of information from the server. For example to check to see if an ID is valid or not.
 - **SQL Select:** For datawindows that are very simple and very unlikely to change, you can use SQL selects. The advantage is that this makes the development very fast. The drawback is that if you need to make any changes to the select logic, you will need to redistribute the

client application (unlike using stored procedures).

- **Error Messages:** All Error conditions should be displayed to the user in a consistent manner.
 - **Return Code Checks:** After every command that accesses the database, you should check the SQLCODE in SQLCA.
 - **Standard Display of Message to User:** Make sure that you display complete and informative messages based on the return codes. This will make the troubleshooting of the application very easy.
- **Report Standards**
 - Report Headers for all reports should be consistently displayed and include the application name, program name of report on top left corner, date, and page number.
 - When creating on-line reports, users should be able to select a Preview Mode to view report on-line and a print option to print report.
 - The application should provide the capability to 'print screen' or print a report if the data on the screen has fields that are scrollable.

Related Policy, Procedures or Standards

see also:

- [Naming Standards](#)
- [Screen Standards](#)
- [Report Standards](#)
- Refer to PowerBuilder and Sybase technical manuals for more information.
- [Accountability & Exceptions](#)

Back to [Programming Standards](#) home page.

Prepared by Andee Korn & Seroj Shirvanian

Last Updated: July 29, 1996